

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平11-353052

(43)公開日 平成11年(1999)12月24日

(51)Int.Cl.<sup>6</sup>G 0 6 F 1/04  
1/32  
9/318

識別記号

3 0 1

F I

G 0 6 F 1/04  
1/00  
9/303 0 1 C  
3 3 2 Z  
3 2 0 B

審査請求 未請求 請求項の数27 OL (全 17 頁)

(21)出願番号

特願平10-138465

(22)出願日

平成10年(1998) 5月20日

(71)出願人 390009531

インターナショナル・ビジネス・マシーンズ・コーポレーション

INTERNATIONAL BUSINESS MACHINES CORPORATION

アメリカ合衆国10504、ニューヨーク州  
アーモンク (番地なし)

(72)発明者 古市 実裕

神奈川県大和市下鶴間1623番地14 日本アイ・ピー・エム株式会社 東京基礎研究所内

(74)代理人 弁理士 坂口 博 (外1名)

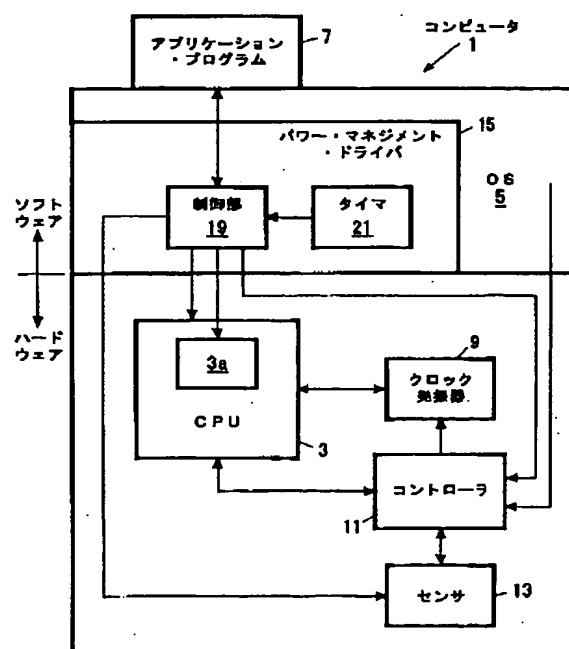
最終頁に続く

(54)【発明の名称】 コンピュータ内のプロセッサの動作速度制御方法及びコンピュータ

(57)【要約】

【課題】単位時間当たりのプロセッサにより実行されたユーザ・モードの命令数及び単位時間当たりのプロセッサの総実行命令数を用いる、プロセッサの動作速度制御方法を提供すること。

【解決手段】コンピュータのパフォーマンス指標の一つとしてユーザ・モードの実行命令数  $I_u$  を、消費電力指標の一つとして総実行命令数  $I_t$  を用いる。これらのパラメータは、エネルギー指標  $E$  と、 $E \propto I_t / I_u$  という関係を有している。パフォーマンス指標の値をユーザ指定の消費電力指標範囲において増大させる、又はエネルギー指標  $E$  の値を減少させるように、CPUの動作速度(動作周波数)を増減する。これにより、省電力とパフォーマンスのバランスをとったパワーマネジメント処理を実施することができる。



1

## 【特許請求の範囲】

【請求項 1】 コンピュータ内のプロセッサの動作速度を制御する方法であって、

第 1 所定期間における、実行されたユーザ・モード命令数  $I_u$  をカウントするステップと、

前記  $I_u$  を参照して、前記プロセッサの動作速度を変更する変更ステップと、を含む動作速度制御方法。

【請求項 2】 前記変更ステップ実施後、再度  $I_u$  をカウントし、再度前記変更ステップを実施することを特徴とする請求項 1 記載の動作速度制御方法。

【請求項 3】 第 2 所定期間における、実行されたユーザ・モード命令数  $I_{u2}$  をカウントするステップと、前記  $I_u$  から前記  $I_{u2}$  への変化率が所定値より大きい場合、前記プロセッサの動作速度を所定値に設定するステップと、

をさらに含む請求項 1 記載の動作速度制御方法。

【請求項 4】 第 2 所定期間における総実行命令数  $I_t$  をカウントするステップをさらに含む、請求項 1 記載の動作速度制御方法。

【請求項 5】 前記  $I_t$  を参照して、前記プロセッサの動作速度を変更する変更ステップをさらに含む請求項 4 記載の動作速度制御方法。

【請求項 6】 前記第 1 所定期間と前記第 2 所定期間が同一であることを特徴とする請求項 4 記載の動作速度制御方法。

【請求項 7】 前記変更ステップが、前記  $I_t$  と  $I_u$  の比を参照して、当該動作速度の変更制御モードを切り換えるステップを含む請求項 6 記載の動作速度制御方法。

【請求項 8】 前記変更ステップが、前記  $I_t$  と  $I_u$  の比を参照して、当該動作速度の変更制御フローを切り換えるステップを含む請求項 6 記載の動作速度制御方法。

【請求項 9】 前記変更ステップ実施後、再度  $I_t$  及び  $I_u$  をカウントし、当該  $I_t$  と  $I_u$  との比を参照して、前記プロセッサの動作周波数を変更するステップをさらに含む請求項 6 記載の動作速度制御方法。

【請求項 10】 コンピュータ内のプロセッサの動作速度を制御する方法であって、

所定期間における、実行されたユーザ・モード命令数  $I_u$  をカウントするステップと、

前記所定期間における総実行命令数  $I_t$  をカウントするステップと、

前記  $I_t$  と  $I_u$  の比を参照して、当該動作速度の変更方針を切り換えるステップと、

前記変更方針を参照して、前記プロセッサの動作速度を変更する変更ステップと、

を含む動作速度制御方法。

【請求項 11】 コンピュータ内のプロセッサの動作速度を制御する方法であって、

2

第 1 所定期間における第 1 パフォーマンス指標の値を測定するステップと、

前記プロセッサの動作速度を変更するステップと、

第 2 所定期間における第 2 パフォーマンス指標の値を測定するステップと、

前記第 1 及び第 2 パフォーマンス指標の値を参照して、

前記プロセッサの動作速度を変更する変更ステップと、

を含む動作速度制御方法。

【請求項 12】 第 3 所定期間において消費電力指標の値を測定するステップと、

前記消費電力指標の値に基づき、前記プロセッサの動作速度を変更するステップとをさらに含む請求項 11 記載の動作速度制御方法。

【請求項 13】 前記変更ステップが、

パフォーマンスが増大するように、前記プロセッサの動作速度を変更するステップを含む請求項 11 記載の動作速度制御方法。

【請求項 14】 コンピュータのプロセッサの動作速度を制御する方法であって、

20 パフォーマンス指標及び消費電力指標の値を測定するステップと、

前記パフォーマンス指標の値と前記消費電力指標の値との比を参照して、プロセッサの動作速度の変更方針を切り換えるステップと、

前記変更方針を参照して、前記プロセッサの動作速度を変更する変更ステップと、

を含む動作速度制御方法。

【請求項 15】 コンピュータのプロセッサの動作速度を制御する方法であって、

30 パフォーマンス指標及び消費電力指標の値を測定するステップと、

前記パフォーマンス指標及び前記消費電力指標の値から、エネルギー指標の値を計算するステップと、

前記エネルギー指標の値を減少させるように、前記プロセッサの動作速度を変更するステップと、

を含む動作速度制御方法。

【請求項 16】 所定期間における、プロセッサにより実行されたユーザ・モード命令数  $I_u$  をカウントする第 1 カウンタと、

40 前記  $I_u$  を参照して、前記プロセッサの動作速度を変更するコントローラと、を有するコンピュータ。

【請求項 17】 前記所定期間における、前記プロセッサの総実行命令数  $I_t$  をカウントする第 2 カウンタをさらに有し、

前記コントローラが、

前記  $I_u$  と  $I_t$  の比を参照して、前記プロセッサの動作速度を変更する請求項 16 記載のコンピュータ。

【請求項 18】 コンピュータのパフォーマンス指標の値を測定する測定器と、

50 前記コンピュータ内のプロセッサの動作速度を変更する

3

コントローラと、  
を有し、  
前記測定器は、第 1 所定期間において第 1 のパフォーマンス指標の値を測定し、  
その後前記コントローラは、前記プロセッサの動作速度を変更し、  
前記測定器は、当該動作速度変更後、第 2 所定期間において第 2 のパフォーマンス指標の値を測定し、  
前記コントローラは、前記第 1 及び第 2 のパフォーマンス指標の値を参照して、前記プロセッサの動作速度を変更するコンピュータ。

【請求項 1 9】前記コントローラは、前記第 1 及び第 2 のパフォーマンス指標の値を参照して、パフォーマンスが増大するように、前記プロセッサの動作速度を変更する請求項 1 8 記載のコンピュータ。

【請求項 2 0】コンピュータのパフォーマンス指標の値を測定する第 1 測定器と、  
前記コンピュータの消費電力指標の値を測定する第 2 測定器と、  
前記パフォーマンス指標の値と前記消費電力指標の値との比を参照して、前記プロセッサの動作速度を変更するコントローラと、  
を有するコンピュータ。

【請求項 2 1】コンピュータのパフォーマンス指標の値を測定する第 1 測定器と、  
前記コンピュータの消費電力指標の値を測定する第 2 測定器と、  
前記パフォーマンス指標及び前記消費電力指標の値から、エネルギー指標の値を計算し、当該エネルギー指標の値を減少させるように、前記プロセッサの動作速度を変更するコントローラと、  
を有するコンピュータ。

【請求項 2 2】コンピュータ内のプロセッサの動作速度を当該プロセッサに変更させるプログラムを格納する記憶媒体であって、  
前記プログラムは、前記プロセッサに、  
所定期間における、実行されたユーザ・モード命令数  $I_u$  のカウントを読み出すステップと、  
前記  $I_u$  を参照して、前記プロセッサの動作速度を設定する設定ステップと、  
を実行させる、記憶媒体。

【請求項 2 3】前記所定期間における総実行命令数  $I_t$  のカウントを読み出すステップをさらに実行させ、  
前記設定ステップは、  
前記  $I_t$  と  $I_u$  の比を参照して、前記動作速度の変更制御フローを切り換えるステップを含む、請求項 2 2 記載の記憶媒体。

【請求項 2 4】コンピュータ内のプロセッサの動作速度を当該プロセッサに変更させるプログラムを格納した記憶媒体であって、

4

前記プログラムは、前記プロセッサに、  
所定期間における、実行されたユーザ・モード命令数  $I_u$  のカウントを読み出すステップと、  
前記所定期間における総実行命令数  $I_t$  のカウントを読み出すステップと、  
前記  $I_t$  と  $I_u$  の比を参照して、前記プロセッサの動作速度を設定するステップと、  
を実行させる、記憶媒体。

【請求項 2 5】コンピュータ内のプロセッサの動作速度を前記プロセッサに変更させるプログラムを格納した記憶媒体であって、

前記プログラムは、前記プロセッサに、  
第 1 所定期間における第 1 パフォーマンス指標の測定結果を読み出すステップと、  
前記プロセッサの動作速度の変更を命ずるステップと、  
第 2 所定期間における第 2 パフォーマンス指標の測定結果を読み出すステップと、  
前記第 1 及び第 2 パフォーマンス指標の値を参照して、  
前記プロセッサの動作速度を設定するステップと、  
を実行させる、記憶媒体。

【請求項 2 6】コンピュータのプロセッサの動作速度を前記プロセッサに変更させるプログラムを格納した記憶媒体であって、  
前記プログラムは、前記プロセッサに、  
パフォーマンス指標及び消費電力指標の測定結果を読み出すステップと、  
前記パフォーマンス指標の値と前記消費電力指標の値との比を参照して、前記プロセッサの動作速度を設定するステップと、  
を実行させる、記憶媒体。

【請求項 2 7】コンピュータのプロセッサの動作速度を前記プロセッサに変更させるプログラムを格納した記憶媒体であって、  
パフォーマンス指標及び消費電力指標の測定結果を読み出すステップと、  
前記パフォーマンス指標及び前記消費電力指標の値から、エネルギー指標の値を計算するステップと、  
前記エネルギー指標の値を減少させるように、前記プロセッサの動作速度を設定するステップと、  
を実行させる、記憶媒体。

【発明の詳細な説明】

【0 0 0 1】

【発明の属する技術分野】本発明は、コンピュータにおける省電力技術に関し、より詳しくは、プロセッサの動作速度（又は動作周波数）を変更する方法に関する。

【0 0 0 2】

【従来の技術】プロセッサの動作速度を減速することにより消費電力を減少させる技術は多数ある。例えば、特開平 9 - 2 3 7 1 3 2 号公報は、CPU の負荷状態、バッテリーの残量、CPU の発熱温度状態に従って、クロッ

50

5

ク周波数の変更を指示し、これらのシステムの状況に応じてクロック周波数を適正值に設定することを開示している。本公報においてCPUの負荷状態は、OSに付属するドライバソフトウェアにより、ある一定時間でのCPU12のアイドル状態の回数から判断される。すなわち、システムコントローラは、CPUがI/O又はメモリをアクセスするごとにビジー状態を示すフラグがセットされるため、このフラグのセット回数に基づいてアイドル状態の回数を認識してCPUの負荷状態を検出する。

【0003】さらに、特開平9-22317号公報（米国特許出願第395335号1995年2月28日出願の対応日本特許出願）は、CPUのアクティビティ及び温度レベルのリアルタイムサンプルに基づいてCPUが休止できるかをモニターが監視し、休止できればハードウェアセクタがCPUのクロック時間を減少し、CPUがアクティブであればCPUを前の高速クロックレベルへ戻す、ポータブルコンピュータ用リアルタイム省電力及び熱管理装置を開示している。モニターはコンピュータの性能レベルを調整しCPUアクティビティ及び温度のリアルタイムサンプリングに回答して省電力及び温度管理を実施する。

【0004】また、特開平9-305569号公報（米国特許出願第010135号1996年1月17日出願の対応特許出願）は、ポータブルコンピュータのCPUの動的動作特性を検出して活動レベルを予測し、電力節約や電力管理を動的に行うため、CPUが第1クロックで動作中に少なくとも1つの動的CPU動作特性を検出し、このCPU動作特性がこの特性に関連する所定の設定点に対して所定の関係を確立するか（設定点割り込み条件が存在するか）判断し、設定点割り込み条件が存在する場合には、第1クロックを変更して所定の設定点を調整することを開示している。また、命令の数を数えてその種類を決定することにより、入出力がほとんどない計算指向モードにCPUが入るかどうかを決定する、という事項も開示している。

【0005】

【発明が解決しようとする課題】しかし、従来技術では、コンピュータのパフォーマンスに関する指標（パフォーマンス指標と呼ぶ）と消費電力に関する指標（消費電力指標と呼ぶ）を参照し、プロセッサの動作速度を制御する全てのアルゴリズムが開示されているわけではない。また、消費電力指標の値に対するパフォーマンス指標の値の比で表されるエネルギー指標を参照する、プロセッサの動作速度制御のアルゴリズムは開示されていない。

【0006】よって、本発明は、パフォーマンス指標を参照してプロセッサの動作速度を制御する新規な方法を提供することが目的である。この際、パフォーマンス指標に加えて消費電力指標を参照するようにしてもよい。

6

【0007】また、エネルギー指標を参照するプロセッサの動作速度制御方法を提供することが目的である。

【0008】さらに、パフォーマンス指標として、単位時間あたりにプロセッサにより実行されたユーザ・モードの命令数を用いる、プロセッサの動作速度制御方法を提供することが目的である。

【0009】加えて、消費電力指標として、単位時間あたりのプロセッサの総実行命令数を用いる、プロセッサの動作速度制御方法を提供することも目的である。

10 【0010】

【課題を解決するための手段】本発明の第1の態様は、コンピュータにおけるプロセッサの動作速度を制御する方法であって、第1所定期間における第1パフォーマンス指標の値を測定するステップと、プロセッサの動作速度を変更するステップと、第2所定期間における第2パフォーマンス指標の値を測定するステップと、第1及び第2パフォーマンス指標の値を参照して、プロセッサの動作速度を変更する変更ステップとを含む。このようにパフォーマンス指標の値を動作速度変更の前後において測定し、その測定結果を用いるようにすれば、パフォーマンス指標の変化傾向に従って、例えば所定の条件でパフォーマンスをなるべく増大させるようにプロセッサの動作速度を変更できる。なお、動作速度は、動作周波数とはほぼ同義であるが、実際の周波数を上下できない場合には、動作期間と非動作期間を設け、その割合を制御することにより、実質的な動作速度又は動作周波数を変更する。

30 【0011】また、第3所定期間において消費電力指標の値を測定するステップと、消費電力指標の値に基づき、プロセッサの動作速度を変更するステップとをさらに含むようにすることも可能である。例えば、消費電力指標の値により動作周波数を設定することも可能である。

【0012】また、本発明の第2の態様は、パフォーマンス指標及び消費電力指標の値を測定するステップと、パフォーマンス指標の値と消費電力指標の値との比を参照して、プロセッサの動作速度の変更方針を切り換えるステップと、変更方針を参照して、プロセッサの動作速度を変更する変更ステップとを含む。先に述べたエネルギー指標又はその逆数を用いて、プロセッサの動作速度を変更するものである。変更方針は、例えばエネルギー指標の変化率が大きい場合に、動作速度変更処理を最初から実施し直す場合、又はエネルギー指標の逆数を所定値にて区分し、それらの区分ごとに定義された動作速度変更処理を実施するようにする場合等がある。

50 【0013】さらに、本発明の第3の態様は、パフォーマンス指標及び消費電力指標の値を測定するステップと、パフォーマンス指標及び消費電力指標の値から、エネルギー指標の値を計算するステップと、エネルギー指標の値を減少させるように、プロセッサの動作速度を変

7

更するステップとを含む。エネルギー指標をより小さくすることは、より省電力で且つパフォーマンスも高い（パフォーマンスの犠牲が小さい）ことを意味するので、より効率的なパワーマネジメントが行われることとなる。

【0014】また、プロセッサのユーザ・モード実行命令数を用いる本発明の第4の態様は、第1所定期間における、実行されたユーザ・モード命令数 $I_u$ をカウントするステップと、 $I_u$ を参照して、プロセッサの動作速度を変更する変更ステップとを含む。ユーザ・モードの実行命令数は、パフォーマンス指標の1つの例であって、ユーザ・モードの実行命令数を多く実行すれば、ユーザから見てタスク・パフォーマンスが高くなる。よって、このユーザ・モード実行命令数を多くするように動作周波数を変更する。但し、動作周波数を上げてもユーザ・モード実行命令数がそれに見合う程常増加するわけではない。

【0015】上記の変更ステップ実施後、再度 $I_u$ をカウントし、再度変更ステップを実施してもよい。これにより、動作速度の変更の効果をフィードバックすることができる。

【0016】また、第2所定期間における、実行されたユーザ・モード命令数 $I_{u2}$ をカウントするステップと、 $I_u$ から $I_{u2}$ への変化率が所定値より大きい場合、プロセッサの動作速度を所定値に設定するステップとを含むようにしてもよい。これにより、プロセッサが実行しているタスクの種類の変更を検出する。

【0017】第2所定期間における総実行命令数 $I_t$ をカウントするステップをさらに含むようにすることもできる。この総実行命令数は、消費電力指標の一つであり、このカウント値を用いて動作速度を制御することも可能である。総実行命令数の割にはユーザ・モード実行命令数が少ない場合もありうる。この場合には、ユーザのタスクはあまり実行されていないことになるので、ユーザから見たタスク・パフォーマンスという点では動作速度の変更の余地がある。

【0018】よって、第1所定期間と第2所定期間を同じにして、 $I_t$ と $I_u$ の比を参照して、動作速度の変更制御モード又は変更制御フローを切り換えるステップを実行してもよい。例えば、 $I_t$ と $I_u$ の比をもって、ユーザ命令指向のモードとシステム命令指向のモードを分けることも、また $I_t$ と $I_u$ の比の変化率等で変化制御フローを切り換えたりすることもできる。

【0019】また、 $I_t$ と $I_u$ の比により動作周波数を変更し、さらに変更後の状態で再度 $I_t$ と $I_u$ をカウントして、動作周波数変更の効果をフィードバックすることもできる。

【0020】また、所定期間における、実行されたユーザ・モード命令数 $I_u$ をカウントするステップと、所定期間における総実行命令数 $I_t$ をカウントするステップ

8

と、 $I_t$ と $I_u$ の比を参照して、プロセッサの動作速度を変更する変更ステップとを含むようにすることもできる。その際には、動作速度の変更方針を切り換えるステップと、変更方針を参照して、プロセッサの動作速度を変更するステップを含むようにすることも可能である。

【0021】以上、本発明の各態様を処理フローの形式で説明したが、それぞれの処理を実施する電子回路や、その他の装置を用いて本発明を実施することもできる。特に、 $I_t$ と $I_u$ をカウントする回路はプロセッサに内蔵される場合もある。また、本発明で必要な各ステップを実施するプログラムを作成することも可能である。このプログラムは、自身で $I_t$ 及び $I_u$ のカウントをせず、専用の回路にてカウントされた値を読み出して、その値に基づき動作速度変更の設定を行うようにすることもできる。プログラムは、記憶装置に記憶されており、流通段階においては、CD-ROMやフロッピー・ディスクなどの記憶媒体に格納されていることもある。

【0022】

【発明の実施の形態】図1に本発明に関連する部分に関するコンピュータ1の構成例を示す。コンピュータ1のハードウェア部分は、CPU (Central Processing Unit) 3と、クロック発振器9と、コントローラ11と、センサ13を含む。なお、センサには、温度センサや、電流計測器、電力計測器等が考えられる。センサ13は、他に必要な情報を取得できる場合には不要である。また、コンピュータ1のソフトウェア部分には、OS (Operating System) 5及びアプリケーション・プログラム7を含み、OS 5にはパワーマネジメント・ドライバ15を含む。このパワーマネジメント・ドライバ15は、制御部19とタイマ21を含んでいる。

【0023】本発明をハードウェアのみで実施する場合、CPU 3に入力されるクロック信号はコントローラ11によって制御される。クロック信号の制御は、CPU 3が対応できる場合には、クロック信号の周波数を変更したり、クロック信号の供給を停止したりすることにより実施する。それらが不可能な場合には、所定のクロック信号を供給しつつ、コントローラ11がCPU 3に所定期間動作の停止を命じることで実施される。また、CPU 3がクロック発振器9のクロック信号から内部クロックを生成している場合には、内部クロック生成の方法を変更することにより、内部クロックの周波数を変更するようにしてもよい。コントローラ11は、CPU 3の動作を観測し、センサ13からの入力を監視する。CPU 3の動作の観測には、例えば、キャッシュミス率や、ノンキャッシュابل・メモリへのアクセス数、I/O命令比率等、単位時間当たりの総実行命令数及びユーザ・モードの実行命令数が考えられる。ユーザ・モードの命令とは、CPU 3の特権レベルが最も低い状態で実行される命令であり、GUIや算術計算など、一般アプリケーション・プログラムの骨格部分を構成する命令の

大部分がこれに相当する。一方、I/O命令などはシステム・モードで実行され、ユーザ・モードとは区別される。また、OS 5からのハードウェア使用に関する情報、OS 5を介して又は介さずに直接、アプリケーション・プログラム 7から得られるハードウェア使用に関する情報等も取得可能である。これらの情報を用いてコントローラ 11は、クロック発振器 9及びCPU 3の動作を制御する。コントローラ 11の動作については後に述べる。なお、コントローラ 11自体をCPU 3内に組み込むことも考えられる。

【0024】本発明をソフトウェアのみで実施する場合、パワー・マネジメント・ドライバ 15の制御部 19が、CPU 3にアクセスして、例えば、単位時間当たりの総実行命令数及びユーザ・モードの実行命令数を検出する。また、センサ 13からその測定結果を取得するようにしてもよい。また、アプリケーション・プログラム 7からハードウェア使用に関する情報を、またOS 5の他の部分からハードウェア使用に関する情報を、得るようにすることも考えられる。制御部 19は、このように取得した情報に基づき、CPU 3の動作速度を制御する。この制御は、コントローラ 11を介してクロック発振器 9の出力クロック信号の制御を行ったり、CPU 3にHalt命令を出力することにより、行われる。例えば、タイマ 21が定期的（例えば244μs）に制御部 19に割り込みをかけ、制御部 19は収集した情報に従い設定された割り合いで、CPU 3にHalt命令を出力するようにすることも可能である。例えば、CPU 3の動作速度を1/4にすると決定した場合には、4回の割り込みのうち、3回はHalt命令を出力し（これによりCPU 3の動作が停止する）、1回は何もしないでCPU 3に処理を実施させるようにする。

【0025】上で述べた、単位時間当たりの総実行命令数及びユーザ・モードの実行命令数は、パーソナル・コンピュータにおいてよく用いられているPentium（Intel社の商標）プロセッサでは、その内部レジスタであるモデル・スペシフィック・レジスタ 3a（Model Specific Register: MSR）から取得可能となっている

（Intel Pentium Processor Family Developer's Manual Vol.1: Pentium Processor, (1995) pp33-1~33-25 参照のこと）。よって、コントローラ 11及び制御部 19は、このレジスタを読みに行けばよい。なお、MSRには累積値が格納されるので、実際にはレジスタ 3aに格納された値の変化量を用いる。単位時間は、MSRの1つの種類であるタイム・スタンプ・レジスタ（Time Stamp Register: TSR）に格納された値を用いることができる。TSRには、CPU 3のクロックごとに1加算された値が格納され、このTSRに格納された値の変化量で、総実行命令数の変化量又はユーザ・モードの実行命令数の変化量を割れば、単位時間当たりの総実行命令数又はユーザ・モードの実行命令数を得ることができる。

【0026】本発明では、コントローラ 11及び制御部 19が行うCPU 3の動作速度制御において、次のような指標を採用する。

P：消費電力指標（プロセッサの消費電力（W））

V：パフォーマンス指標（単位時間あたりに処理したタスクの問題サイズ（計算速度））

T：与えられた問題を処理するのに要した時間（ $V = 1/T$ ）

E：エネルギー指標

10 エネルギー指標は、 $E = P/V$ （又は $E \propto P/V$ ）で表される。一般に、Pが大きくなると、Vも増加する。すなわち、消費電力が増加するように高速でCPU 3を動作させると、処理速度も早くなる。しかし、個々のタスクごとに、I/O命令の比率や、キャッシュミス率等が異なり、タスクによっては、Pの増加に対してVが徐々に飽和してCPU 3の動作速度の増加が必ずしもパフォーマンスの上昇にはつながらない場合もある。この場合には、パフォーマンスと省電力の両方の観点からエネルギー指標Eをより小さくするような速度でCPUを動作させれば、より効率的である。一方、ユーザによって、20 所定の消費電力範囲内でパフォーマンスをより増大させることを欲する場合もある。このような場合には、パフォーマンス指標Vがより高くなるように制御する。

【0027】なお、消費電力指標として用いることができる情報としては、所定時間内の総実行命令数、センサ 13が温度センサである場合には温度から計算される熱量、センサ 13が電流計測器である場合には測定された電流値、センサ 13が電力計測器である場合には測定された電力値、等が考えられる。また、パフォーマンス指標としては、所定時間内のユーザ・モードの実行命令数、30 キャッシュミス率、ノンキャッシュブルメモリへのアクセス数、I/O命令比率、OS 5又はアプリケーション・プログラム 7からのハードウェア使用に関する情報（例えば、タスクごとのCPU占有率、プロセスの優先度、アプリケーション・プログラムの要求、又はそれらの組み合わせ）が考えられる。エネルギー指標は、上述の式から得られる。

【0028】以上のようなパフォーマンス指標及び電力指標の値を取得し、それらを用いて目標を達成するような制御フローを以下に述べる。

【0029】（1）エネルギー指標Eを減少させるための処理フロー

（a）第1例

エネルギー指標Eの最小化を目標とする第1例の処理フローを図2に示す。最初に、CPU 3の動作速度f<sub>i</sub>（iは図2における繰り返しの回数を示す）を最大値f<sub>max</sub>に設定する（ステップ103）。そして、その時のエネルギー指標E（f<sub>i</sub>）を取得（測定）する（ステップ105）。先に述べたとおり、パフォーマンス指標V及び消費電力指標Pを測定して、計算にて取得する。も

11

し、 $|E(f_i) - E(f_{i-1})| / \Delta T$  が所定のしきい値  $E_0$  を超えている場合（エネルギー指標  $E$  の変化量がしきい値  $E_0$  を超えている場合）には、CPU が実行しているタスクの性質が切り替わったと判断して、処理をステップ 103 に戻す。このようにエネルギー指標  $E$  を参照して、CPU 3 の動作速度の変更処理フローを切り換える。なお、 $\Delta T$  は図 2 におけるステップ 105 から次のステップ 105 までのループの実行間隔であって、例えば数百  $\mu s$  程度である。

【0030】タスクの性質が切り替わっていないようであれば（図 2 の最初の処理は  $f_{i-1}$  がないので必ず切り替わっていないとする）、CPU 3 の動作速度を  $\Delta t$  の間だけ、 $f_- = f_i - \Delta f_0$  に設定する（動作速度を  $\Delta f_0$  だけ下げる。ステップ 109）。 $\Delta t$  は、一時的に CPU 3 の動作速度を調整するための期間であるから、例えば数十  $\mu s$  程度である。そして、再度エネルギー指標  $E(f_-)$  を取得し（ステップ 111）、 $E(f_-) < E(f_i)$  が成り立つかどうか判断する（ステップ 113）。もし、成り立つようであれば、CPU 3 の動作速度を減少させた方がよいので、既に  $f_i$  が  $f_{\min}$  でなければ、 $f_{i+1} = f_i - \Delta f_1$  を設定する（ $\Delta f_1$  だけ減速する。ステップ 117）。一方、ステップ 113 の条件が成り立たない場合には、CPU 3 の動作速度を減少させるとエネルギー指標  $E$  に悪影響を及ぼすので、既に  $f_i$  が  $f_{\max}$  でなければ、 $f_{i+1} = f_i + \Delta f_1$  を設定する（ $\Delta f_1$  だけ高速化する。ステップ 115）。このように第 1 例では、 $E$  の値を参照して、CPU 3 の動作速度の変更方針を決定している。なお、本例では  $\Delta f_1$  は  $\Delta f_0$  より大きな値とするが、同じでも問題ない。そして、 $i$  を 1 インクリメントしてステップ 105 に戻る（ステップ 119）。

【0031】この処理を繰り返し実施することにより、エネルギー指標  $E$  を最小化するように CPU 3 の動作速度（又は駆動周波数） $f$  を変更する。

【0032】（2）第 2 例

図 3 にエネルギー指標  $E$  を減少させるための処理フローを示す。本処理フローでは、エネルギー指標  $E$  は最低にはならないが、エネルギー指標  $E$  をある許容範囲内に収めるようになっている。なお、第 2 例では  $V/P = 1/R$  という値を導入する。単位時間当たりの総実行命令数  $I_t$  及びユーザ・モードの実行命令数  $I_u$  を用いて処理を実行する場合には、 $R$  は、ユーザ・モードの実行命令数の割合を示しており、所定のしきい値  $R_{1H}$  より高い場合には、CPU 3 はユーザ命令指向の状態にあると言える。また、 $R$  が所定のしきい値  $R_{1L}$  より低い場合には、CPU 3 はシステム命令指向の状態にあると言える。

【0033】まず、CPU 3 の動作速度  $f$  を  $f_{\max}$  に設定する（ステップ 123）。そして、 $f_{\max}$  の時の  $R$  の値  $R_0$  が所定のしきい値  $R_{1H}$  を超えているかどうか判断

12

する（ステップ 125）。もし、超えていない場合には、 $R_0$  が所定のしきい値  $R_{1L}$  より小さいかどうか判断する（ステップ 135）。もし、 $R_0$  が所定のしきい値  $R_{1L}$  以上である場合には、 $R_0$  は  $R_{1L} \leq R_0 \leq R_{1H}$  であって、エネルギー指標  $E$  は許容範囲内に収まっており、これ以上の制御は必要ないとして、ステップ 133 に移行する。

【0034】一方、 $R_0$  が  $R_{1H}$  より大きい場合には、CPU 3 の動作速度  $f$  を  $\Delta f$  だけ減速する（ステップ 127）。もし、減速した後の動作速度が最低の動作速度である場合には、ステップ 133 に移行する（ステップ 129）。そうでない場合には、減速した後の動作速度で、パフォーマンス指標  $V$  及び消費電力指標  $P$  を測定し、 $V > V_0 * R_3$  且つ  $V/P > R_{1H}$  であるかどうか判断する（ステップ 131）。第 1 の式は、CPU 3 の動作速度を減速した際に下がるパフォーマンス指標  $V$  の許容範囲を定めるものであって、動作速度が  $f_{\max}$  であった時のパフォーマンス指標値  $V_0$  に対する割合である。また第 2 の式は、CPU 3 の動作速度を減速した後に  $R = V/P$  がステップ 125 におけるしきい値  $R_{1H}$  の範囲内にとどまっているかどうかを判断するためのものである。単位時間当たりの総実行命令数  $I_t$  及びユーザ・モードの実行命令数  $I_u$  を用いて処理を実行する場合には、動作速度減速後もユーザ命令指向の状態にあるかどうかを判断している。もし、第 1 及び第 2 の式を満たしている場合には、さらに CPU 3 の動作速度を減速する。一方、第 1 又は第 2 の式を満たしていない場合には、その時の動作速度に固定してステップ 133 に移行する。

【0035】また、 $R_0 < R_{1L}$  を満たす場合（ステップ 135）には、CPU 3 の動作速度  $f$  を  $\Delta f$  だけ減速する（ステップ 137）。もし、減速した後の動作速度が最低の動作速度である場合には、ステップ 133 に移行する（ステップ 129）。そうでない場合には、減速した後の動作速度で、パフォーマンス指標  $V$  及び消費電力指標  $P$  を測定し、 $V > V_0 * R_2$  且つ  $V/P < R_{1L}$  であるかどうか判断する（ステップ 131）。第 1 の式は、CPU 3 の動作速度を減速した際にパフォーマンス指標  $V$  の許容範囲を定めるものであって、動作速度が  $f_{\max}$  であった時のパフォーマンス指標  $V_0$  に対する割合である。また第 2 の式は、CPU 3 の動作速度を減速した後に  $R = V/P$  がステップ 135 におけるしきい値  $R_{1L}$  の範囲内にとどまっているかどうかを判断するためのものである。単位時間当たりの総実行命令数  $I_t$  及びユーザ・モードの実行命令数  $I_u$  を用いて処理を実行する場合には、動作速度減速後もシステム命令指向の状態にあるかどうかを判断している。もし、第 1 及び第 2 の式を満たしている場合には、さらに CPU 3 の動作速度を減速する。一方、第 1 又は第 2 の式を満たしていない場合には、その時の動作速度に固定してステップ 133 に移行

する。

【0036】以上のように、第2例では $R_{1H}$ 及び $R_{1L}$ を用いて、CPU3の動作速度の変更処理モード(変更処理方針)を設定し、それに基づき処理を実施している。ステップ127及びステップ137の $\Delta f$ は同じでも異なる値でもよい。

【0037】ステップ133では、パフォーマンス指標 $V$ の変化率の絶対値を検査する。すなわち、パフォーマンス指標 $V$ の変化率の絶対値が所定のしきい値 $R4$ を超えたかどうか判断する。この処理は定期的又は必要に応じて行なわれ、この条件が満たされないうちは、同じ動作速度でCPU3を動作させる。この条件が満たされた時には、CPU3で処理されるタスクの種類が変更されたとしてステップ123に戻る。パフォーマンス指標 $V$ により処理フローが変更されたと考えることもできる。

【0038】 $R_{1H}$ 及び $R_{1L}$ は同じ値でもよい。また、 $R2$ 及び $R3$ も同一でもよい。これらの定数及び $R4$ は、固定にしてもよいし、場合によっては動的に変更することも可能である。

【0039】第1及び第2例ともエネルギー指標 $E$ (より一般的には、パフォーマンス指標と消費電力指標 $P$ の比)に着目して行われる処理フローである。

【0040】(2)パフォーマンス指標 $V$ を増大させるための処理フロー

(a) 第1例

本例はエネルギー指標 $E$ を用いない。但し、消費電力指標 $P$ を用いてCPU3の動作速度を変更する処理を含む。なお、CPU3の動作速度の初期値 $P(f_1)$ は設定できる最大値 $f_{max}$ に設定されている。まず、動作速度 $f_i$ における消費電力指標 $P(f_i)$ を取得する(ステップ153)。 $f$ のサフィックスは、図4の処理の繰り返し数を表す。そして、 $P(f_i)$ が、ユーザ指定の $P_{min}$ 及び $P_{max}$ の間にあるかどうか判断する(ステップ155)。もし、 $P_{max}$ 以上である場合、又は $P_{min}$ 以下である場合には、 $f_i$ への変更が不適当であった、又は $f_i$ 設定後にタスクの性質が切り替わった場合であるから、一旦 $f_i$ を $f_{i-1}$ (前回図4の処理を開始する際の動作速度)に戻す(ステップ157)。

【0041】ステップ155又はステップ157の後、 $f_i$ におけるパフォーマンス指標 $V(f_i)$ を取得する(ステップ159)。そして、CPU3の動作速度を $\Delta t$ の間だけ $\Delta f_0$ 上昇させる(ステップ161)。すなわち、 $f_+=f_i+\Delta f_0$ と設定する。その後、 $f_+$ におけるパフォーマンス指標 $V(f_+)$ を取得する(ステップ163)。もし、 $V(f_+) > V(f_i)$ ならば(ステップ165)、CPU3の動作速度を上昇させた方がパフォーマンス指標 $V$ の値が上昇するので、 $f_{i+1}=f_i+\Delta f_1$ と設定する(ステップ177)。一方、 $V(f_+) \leq V(f_i)$ であるならば、CPU3の動作速度を $\Delta t$ の間だけ

だけ $\Delta f_0$ 減速する(ステップ167)。すなわち、 $f_-=f_i-\Delta f_0$ を設定する。そして、 $f_-$ におけるパフォーマンス指標 $V(f_-)$ を取得する(ステップ169)。その後、 $V(f_-)$

$V(f_i)$ であるか判断する(ステップ171)。この条件が満たされないということは、CPU3の動作速度を変更しなくともパフォーマンスは変わらないので、 $f_{i+1}=f_i$ に設定する(ステップ173)。一方、 $V(f_-)$

10  $V(f_i)$ でない場合には、 $f_{i+1}=f_i-\Delta f_1$ を設定する(ステップ175)。

【0042】以上の処理を所定時間間隔又は必要に応じて実施する。ユーザ指定の省電力指標に従い、省電力とパフォーマンスの両方のバランスをとることができる。なお、第1例では $\Delta f_0$ と $\Delta f_1$ は、それぞれ $\Delta f_1$ と $\Delta f_1$ より小さいものとして考えているが、同じ値であってもよい。また、上付サフィックス+及び-は異なる値でも同じ値でもよい。

【0043】(b) 第2例

20 図5に、パフォーマンス指標 $V$ を増大させるための処理フローの第2例を示す。第2例では、パフォーマンス指標 $V$ の変化率を考慮して、動作速度変更の処理フローを切り換える。なお、図5に示されている処理は、第*i*回目の繰り返し処理である。この処理は、所定間隔で又は必要に応じて繰り返し実行される。なお、CPU3の動作速度の初期値 $P(f_1)$ は設定できる最大値 $f_{max}$ に設定されている。

30 【0044】まず、設定されているCPU3の動作速度 $f_i$ における消費電力指標 $P(f_i)$ を取得する(ステップ183)。そして、ユーザ指定の $P_{max}$ と比較する(ステップ185)。もし、 $P_{max}$ より $P(f_i)$ が大きい場合には、消費電力を減らさずべく、 $f_{i+1}=f_i-\Delta f_1$ を設定する(ステップ189)。また、ユーザ指定の $P_{min}$ と $P(f_i)$ を比較し(ステップ187)、もし $P_{min}$ より小さければ、パフォーマンス指標の値を上げるためにCPU3の動作周波数を上げることができるので、 $f_{i+1}=f_i+\Delta f_1$ を設定する(ステップ191)。

40 【0045】ステップ185及びステップ187の条件を両方満たしていない場合には、CPU3の動作速度を制御できる。よって、 $f_i$ におけるパフォーマンス指標 $V(f_i)$ を取得する(ステップ193)。次に、CPU3の動作速度を $\Delta t$ の間だけ $\Delta f_1$ 減速する(ステップ195)。よって、 $f_-=f_i-\Delta f_0$ を設定する。そして、 $f_-$ におけるパフォーマンス指標 $V(f_-)$ を取得する(ステップ197)。取得したパフォーマンス指標を用いて、 $V(f_-)$

50  $V(f_i)$ であるかどうか判断する(ステップ199)。もし、この条件が満たされないならば、CPU3の動作速度を変更する必要がないので、 $f_{i+1}=f_i$ と設定す



15

る（ステップ201）。一方、ステップ199の条件を満たす場合には、 $f_{i+1} = f_i - \Delta f_1$ を設定する（ステップ203）。そして、次に、パフォーマンス指標Vの変化率を検査する。これは、 $| (V(f_i) - V(f_{i-1})) / \Delta t |$ で計算する。これが所定の $V_0$ を超えている場合には、CPU3が処理しているタスクの種類が切り換えられたとして、 $f_{i+1}$ を設定できる最大値 $f_{max}$ に設定する（ステップ207）。タスクの種類が切り換えられていない場合には、ステップ203又はステップ201において設定されたままで、処理を終了する（ステップ209）。

【0046】以上の処理を実行することにより、ユーザ指定の消費電力指標とパフォーマンスの両方のバランスをとることができる。なお、第2例では $\Delta f_0$ は $\Delta f_1$ より小さいと考えているが、同じ値であってもよい。

【0047】（3）第3例

本例は、図5を発展させたアルゴリズムである。まず、 $J = (dV/dP) / (V/P)$ という、 $V/P$ で正規化された指標Jを導入する。このJの最小値である $J_{min}$ をさらに規定しておく。このJが1.0より大きい又は1.0程度の値である場合には、より多くの電力が供給されるとJの分子は減少せずにJ自体も増加する。よって、より多くの電力を供給した時にパフォーマンスに改善の余地があるということを示す。一方、このJが1.0より小さい（ $J < J_{min}$ ）ならば、より多くの電力が供給されてもJの分子は減少し、パフォーマンスの改善にはつながらないということを示している。

【0048】さらに、タスク特性の変化を検出するために、 $S = |V_i/P_i - V_0/P_0| / (V_0/P_0)$ という指標Sを導入する。ここで $V_i$ 及び $P_i$ は、それぞれその初期値である。 $V_i$ 及び $P_i$ は、それぞれ最新のV及びPである。タスク特性の変化を検出するためのしきい値 $S_{max}$ も規定しておく。

【0049】加えて、プロセッサの動作速度を最高にした時の各タスクの実行数をカウントするための指標Countも導入する。これは、タスクが安定状態になる前にプロセッサの動作速度を変えてしまったり、動作速度の変更をあまりに多くしないようにするために用いられる。よって、タスクが安定状態になったということを確認するためのしきい値をCとする。

【0050】図6にこれらの指標を用いたプロセッサの動作速度調整アルゴリズムの一例を示す。このアルゴリズムは図6に示すように、タスクごとに行われ、各変数は各タスクごとに保存される。まず、このタスクAが新たなタスクであるかどうか判断される（ステップ221）。もし、新たなタスクであれば、処理状態を表わす変数Searchをstart状態に設定し、先に述べたCountを初期値0に設定し、プロセッサの動作速度Speedを最高速度MAXに設定する（ステップ223）。もし、タスクAが新たなタスクでない場合には、前回タスクAについて

16

決定された変数値を取り出す。そして、プロセッサの動作速度をSpeedに設定し、P及びVの測定を開始する（ステップ225）。そして、タスクAを実行する（ステップ227）。もし、タスクAが新たなタスクであれば、ここで取得されるP及びVは、 $P_0$ 及び $V_0$ となり、次のタスクにスイッチする。

【0051】その後、プロセッサ速度をその最高速度MAXにセットし、その時の $V_i$ 及び $P_i$ （ $V_{count+1}$ 及び $P_{count+1}$ ）を測定する。さらに、Countを1インクリメントする（ステップ229）。そして、 $S < S_{max}$ であるかどうか判断する（ステップ231）。もし、 $S < S_{max}$ でなければ、タスク特性が変わっているので、変数Searchをstart状態にセットし、countを初期値0に戻し、プロセッサの動作速度Speedを最高速度MAXにセットする（ステップ233）。一方、タスク特性が変わっていなければ、現在のCount値がしきい値Cより大きいと判断する（ステップ235）。しきい値CよりCount値が小さければ、またタスクは安定状態でないので、その後の処理を行わず、次のタスクにスイッチする（ステップ245）。もし、タスクが安定状態であれば、処理状態を表わす変数SearchがEndを表わしているか検査する（ステップ237）。このEndは、プロセッサ動作速度の調整が済んでいることを表わす。よって、Search=Endであれば、次のタスクにスイッチする。

【0052】そうでなければ、 $J < J_{min}$ であるか判断する（ステップ239）。この条件に合致しない場合には、プロセッサの動作速度を減少させることができるので、Speedを一段階（ $\Delta f$ ）減速するようにセットする（ステップ243）。すなわち、 $Speed = Speed - 1$ とする。そして、次のタスクにスイッチする。一方、ステップ239の条件を満たしていない場合には、これ以上減速できないので、一回前のプロセッサ動作速度に戻し（ $Speed = Speed + 1$ ）、処理状態Searchを終了状態Endに変更する（ステップ241）。この後、次のタスクにスイッチする。

【0053】以上の処理は、タスクのスイッチを行うOSのスケジューラが行う。なお、上で述べたパワーマネジメント・ドライバと協調してもよい。

【0054】以上のような処理をコントローラ11又はパワーマネジメント・ドライバ15内の制御部19により実施する。なお、図1のブロック図は一例であって、例えばタイマ21はパワーマネジメント・ドライバ15内に設けているが、OS5の他の部分にて有するタイマを用いても、ハードウェアのタイマを使用してもよい。さらに、センサ自体をCPU13が含むようにすることも可能である。本発明のコントローラ11は、他の周辺回路と一緒にチップに実装しても別個に実装してもよい。図1ではアプリケーション・プログラム7は1つしか描かれていないが、当然複数実行されている場合もある。なお、図1では機能ブロック間の参照のための

17

線を必要と考えられる部分のみ描いているが、機能間で図 1 に描かれていない参照を設けることも可能である。

【0055】また図 2 乃至図 6 に示したアルゴリズムは一例であって、CPU 3 の動作速度を目的に応じて調整できるようなアルゴリズムであれば、図 2 乃至図 6 と異なってもよい。また、OS 5 に他のパワーマネジメント機能（例えば APM ドライバ (Advanced Power Management)）が含まれている場合には、それらとの整合をとる必要がある場合も生じ得る。

【0056】

【効果】パフォーマンス指標を参照してプロセッサの動作速度を制御する新規な方法を提供することができた。また、消費電力指標も参照するような方法も提供できた。

【0057】また、エネルギー指標を参照するプロセッサの動作速度制御方法を提供することができた。

【0058】さらに、パフォーマンス指標として、所定期間においてプロセッサにより実行されたユーザ・モードの命令数を用いる、プロセッサの動作速度制御方法を提供することができた。

【0059】加えて、消費電力指標として、所定期間におけるプロセッサの総実行命令数を用いる、プロセッサの動作速度制御方法を提供することもできた。

【図面の簡単な説明】

18

【図 1】本発明の装置構成例を示したブロック図である。

【図 2】エネルギー指標を減少させるための第 1 処理フローを示した図である。

【図 3】エネルギー指標を減少させるための第 2 処理フローを示した図である。

【図 4】パフォーマンス指標を増大させるための第 1 処理フローを示した図である。

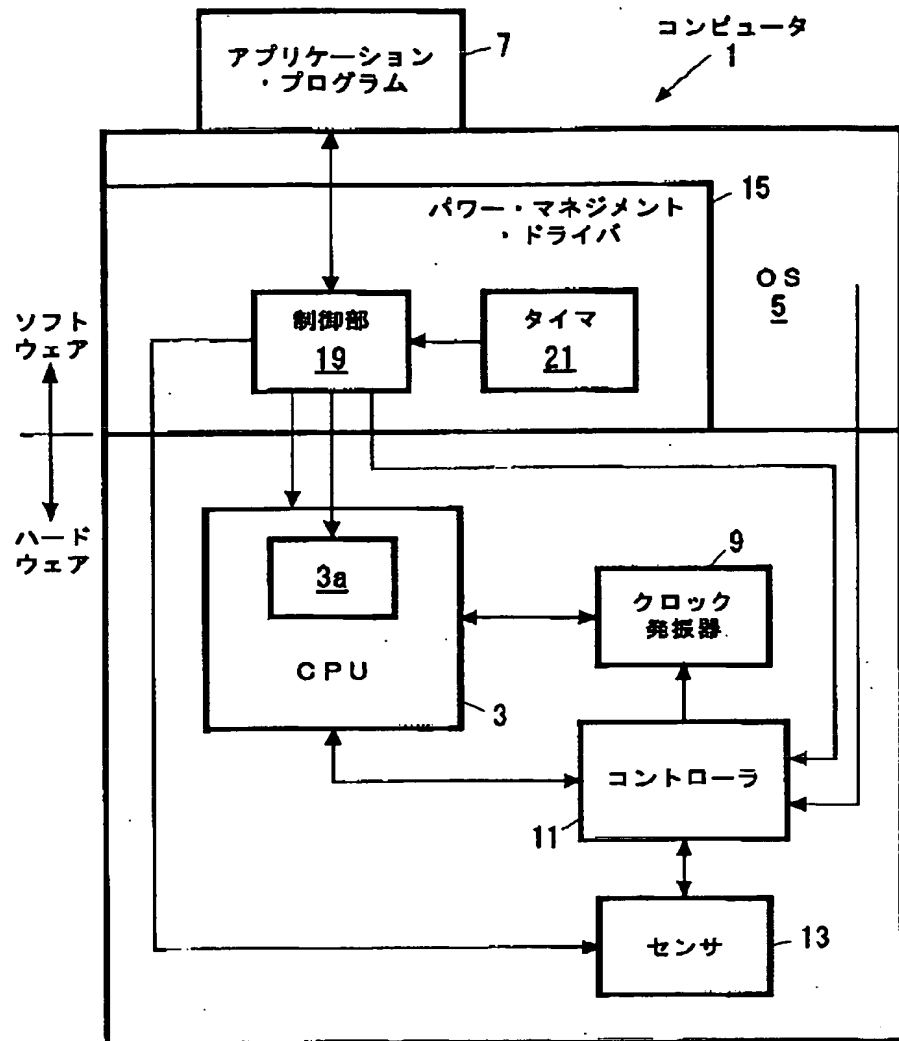
10 【図 5】パフォーマンス指標を増大させるための第 2 処理フローを示した図である。

【図 6】パフォーマンス指標を増大させるための第 3 処理フローを示した図である。

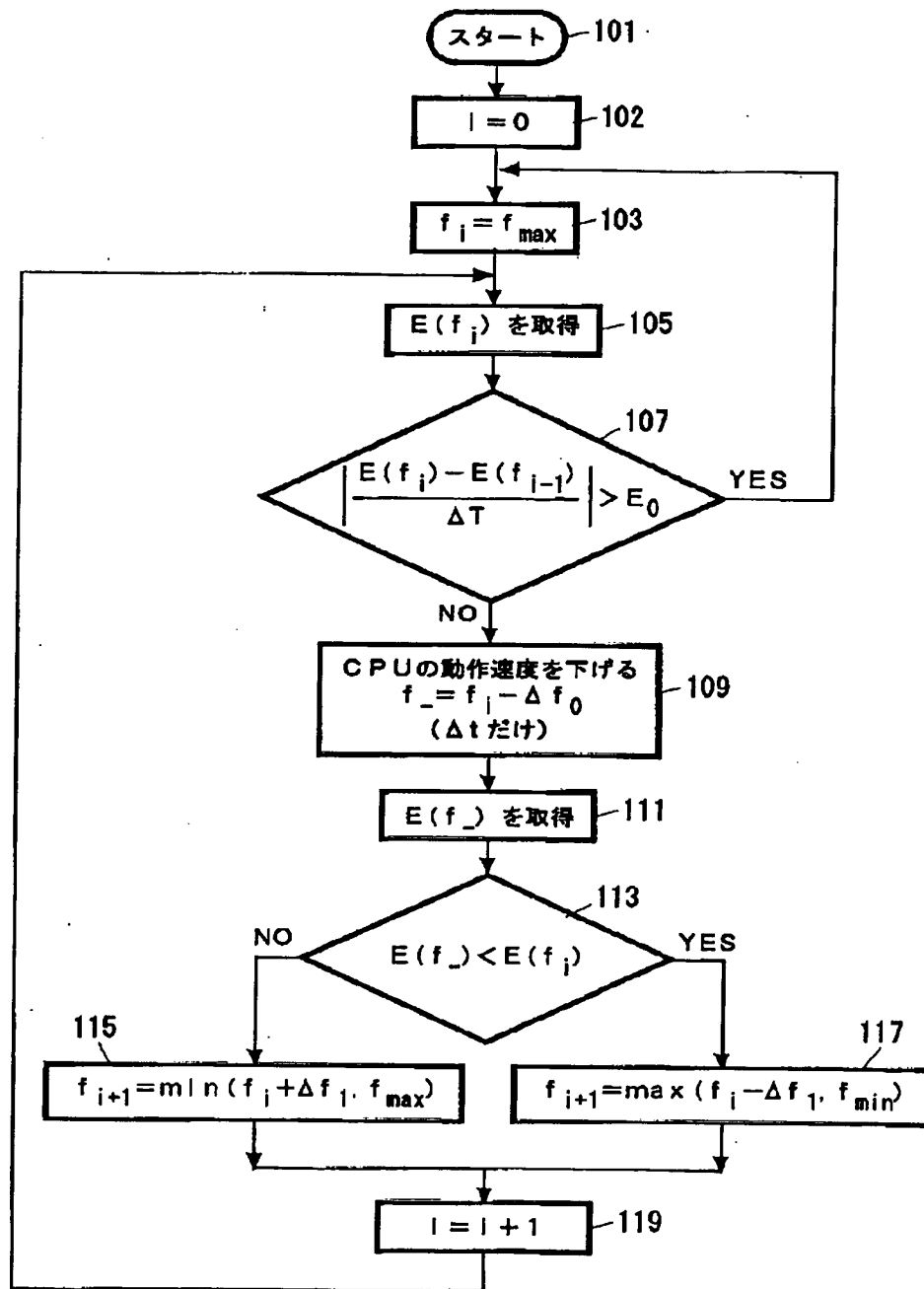
【符号の説明】

- 1 コンピュータ
- 3 CPU
- 3a MSR
- 5 OS
- 7 アプリケーション・プログラム
- 9 クロック発振器
- 20 11 コントローラ
- 13 センサ
- 15 パワーマネジメント・ドライバ
- 19 制御部
- 21 タイマ

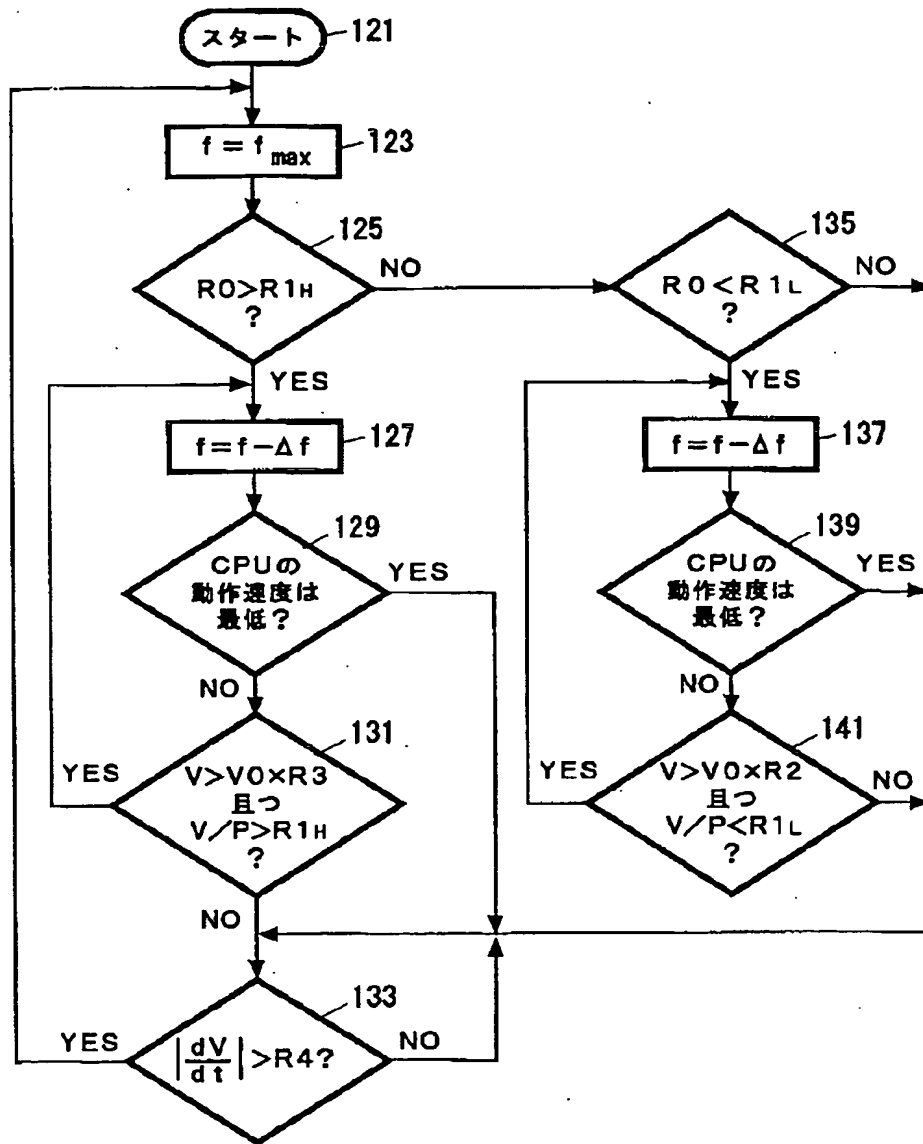
【図 1】



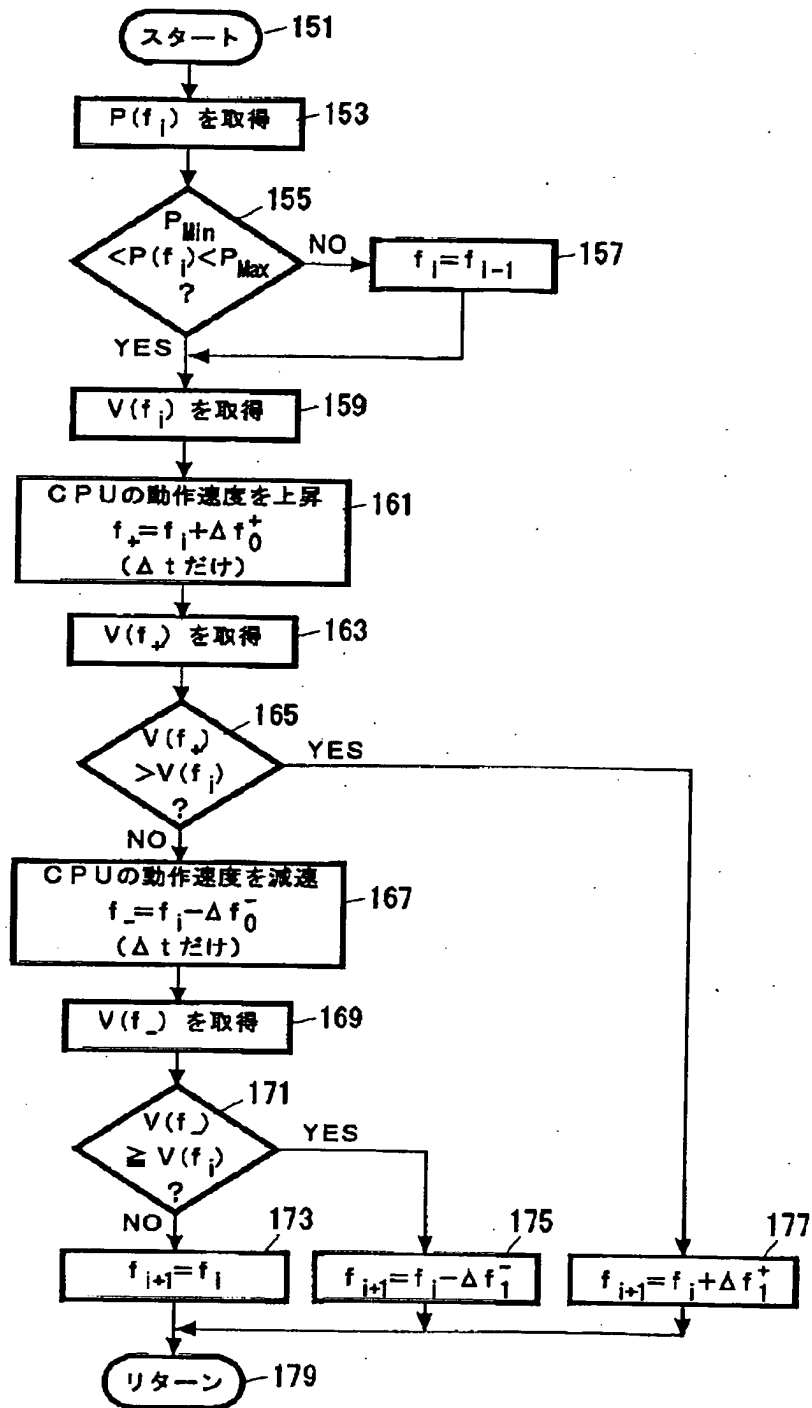
【図2】



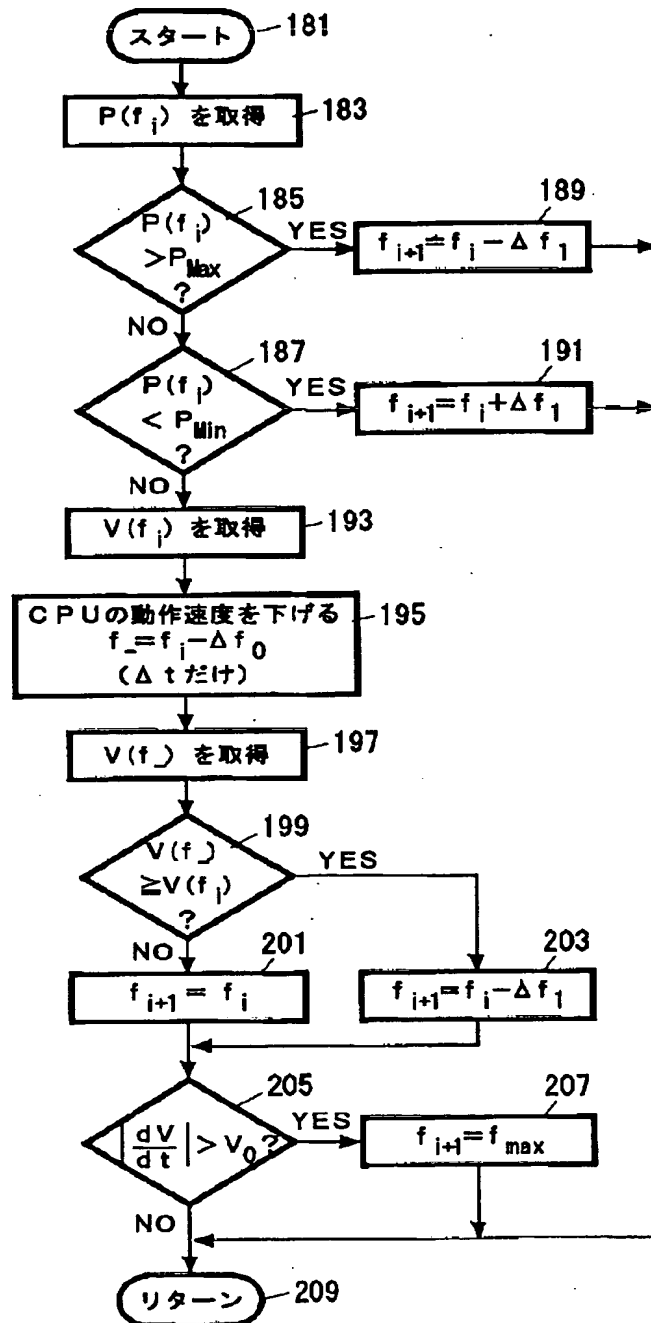
【図 3】



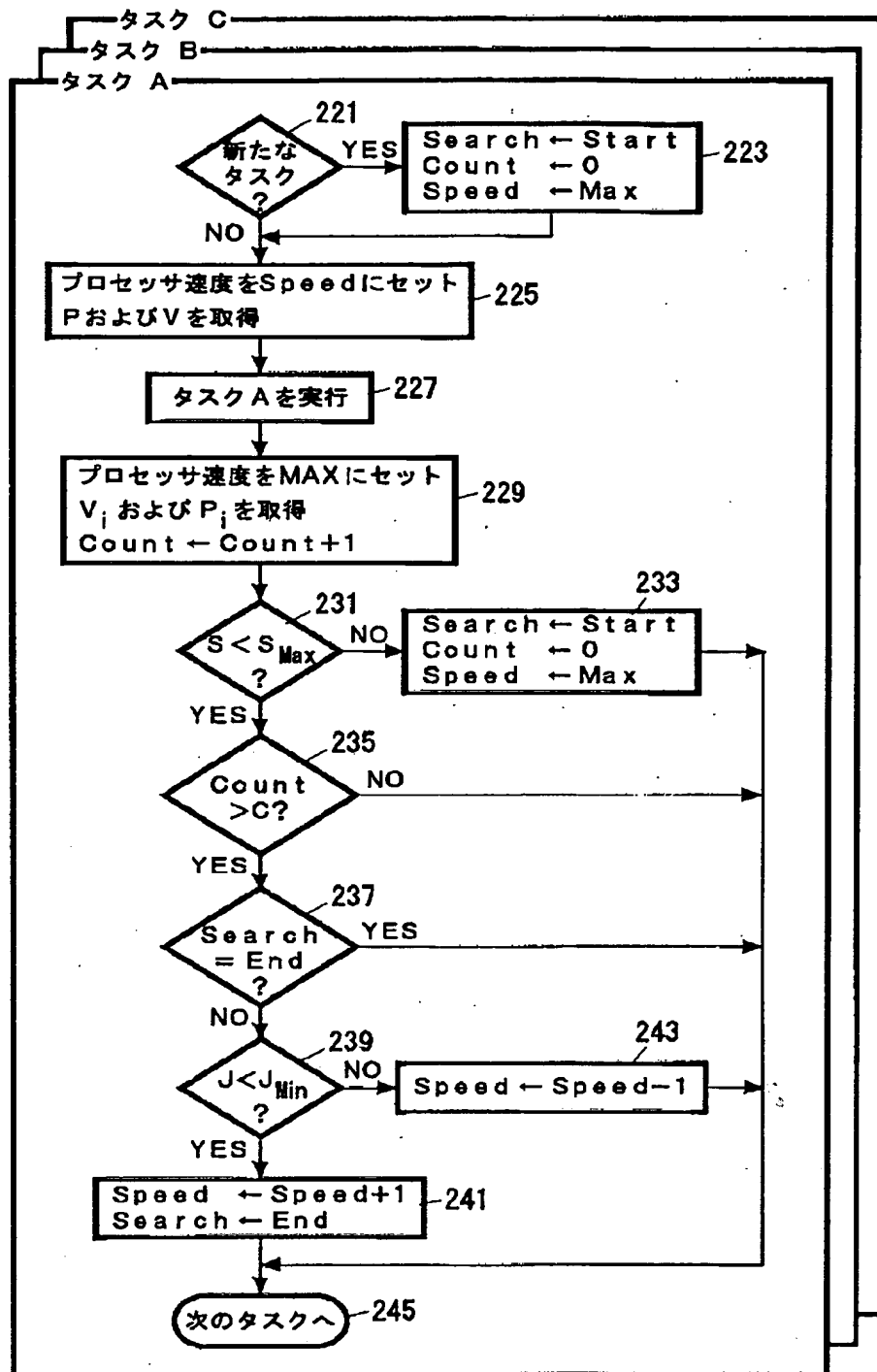
【図 4】



【図 5】



【図6】





## 【手続補正書】

【提出日】平成 11 年 1 月 29 日

## 【手続補正 1】

【補正対象書類名】明細書

【補正対象項目名】0041

【補正方法】変更

## 【補正内容】

【0041】ステップ 155 又はステップ 157 の後、 $f_i$  におけるパフォーマンス指標  $V(f_i)$  を取得する（ステップ 159）。そして、CPU3 の動作速度を  $\Delta t$  の間だけ  $\Delta f_0$  上昇させる（ステップ 161）。すなわち、 $f_+ = f_i + \Delta f_0$  と設定する。その後、 $f_+$  におけるパフォーマンス指標  $V(f_+)$  を取得する（ステップ 163）。もし、 $V(f_+) > V(f_i)$  ならば（ステップ 165）、CPU3 の動作速度を上昇させた方がパフォーマンス指標  $V$  の値が上昇するので、 $f_{i+1} = f_i + \Delta f_1$  と設定する（ステップ 177）。一方、 $V(f_+) \leq V(f_i)$  であるならば、CPU3 の動作速度を  $\Delta t$  の間だけ  $\Delta f_0$  減速する（ステップ 167）。すなわち、 $f_- = f_i - \Delta f_0$  を設定する。そして、 $f_-$  におけるパフォーマンス指標  $V(f_-)$  を取得する（ステップ 169）。その後、 $V(f_-) \geq V(f_i)$  であるか判断する（ステップ 171）。この条件が満たされないということは、CPU3 の動作速度を変更しなくともパフォーマンスは変わらないので、 $f_{i+1} = f_i$  に設定する（ステップ 173）。一方、 $V(f_-) \geq V(f_i)$  でない場合には、 $f_{i+1} = f_i - \Delta f_1$  を設定する（ステップ 175）。

## 【手続補正 2】

【補正対象書類名】明細書

【補正対象項目名】0045

【補正方法】変更

## 【補正内容】

【0045】ステップ 185 及びステップ 187 の条件を両方満たしていない場合には、CPU3 の動作速度を制御できる。よって、 $f_i$  におけるパフォーマンス指標  $V(f_i)$  を取得する（ステップ 193）。次に、CPU3 の動作速度を  $\Delta t$  の間だけ  $\Delta f_1$  減速する（ステップ 195）。よって、 $f_- = f_i - \Delta f_0$  を設定する。そして、 $f_-$  におけるパフォーマンス指標  $V(f_-)$  を取得する（ステップ 197）。取得したパフォーマンス指標を用いて、 $V(f_-) \geq V(f_i)$  であるかどうか判断する（ステップ 199）。もし、この条件が満たされないならば、CPU3 の動作速度を変更する必要がないので、 $f_{i+1} = f_i$  と設定する（ステップ 201）。一方、ステップ 199 の条件を満たす場合には、 $f_{i+1} = f_i - \Delta f_1$  を設定する（ステップ 203）。そして、次に、パフォーマンス指標  $V$  の変化率を検査する。これは、 $|V(f_i) - V(f_{i-1})| / \Delta t$  で計算する。これが所定の  $V_0$  を超えている場合には、CPU3 が処理しているタスクの種類が切り換えられたとして、 $f_{i+1}$  を設定できる最大値  $f_{\max}$  に設定する（ステップ 207）。タスクの種類が切り換えられていない場合には、ステップ 203 又はステップ 201 において設定されたまま、処理を終了する（ステップ 209）。

フロントページの続き

(72)発明者 相原 達  
神奈川県大和市下鶴間1623番地14 日本ア  
イ・ビー・エム株式会社 東京基礎研究所  
内

(72)発明者 下遠野 享  
神奈川県大和市下鶴間1623番地14 日本ア  
イ・ビー・エム株式会社 東京基礎研究所  
内

**THIS PAGE BLANK (USPTO)**